Smart Grid LAB Hessen

WHITE PAPER

Smart Grid AI: Energy Forecast with Machine Learning

JUNE 22, 2022

TRACTEBEL ENGINEERING GMBH

DR. LUCA PIZZIMBONE

EUROPEAN UNION
Investing in Your Future
European Regional Development Fund

HESSEN
Ministry for Economic Affairs, Energy, Transport and Housing
State of Hessen

In our previous article "Machine Learning in Smart Grid Application ", we discussed how machine learning can support smart grid applications, providing an introduction to Artificial Intelligence (AI), classical machine learning and deep learning. We looked at the current and most promising applications of these technologies and we discussed the results of an energy forecast use case.

This article provides further insights on the same use case, presenting the architectural decisions, the analysis process and some additional information on feature engineering and model evaluation used in the study.

1 Introduction

About ten years back, the UK government wanted energy providers to install smart meters in every home in England, Wales and Scotland. There were more than 26 million homes, with the goal of every home having a smart meter by last year (2020).

The dataset we used in this analysis is a refactored version of the data from the London data store, which contains the energy consumption readings for about two years for a sample of 5,567 London Households that took part in the UK Power Networks led Low Carbon London project.

The choice of the best machine learning algorithm is not always obvious. Sometime the use of classical simple machine learning models is just fine and there is no need to increase complexity and computational costs. However, there are challenging situations, where to achieve the desired accuracy, is necessary to use more complex models.

Classical machine learning and deep-learning models have been defined with the aim to compare their performance on the prediction of daily energy consumption and to achieve an energy forecast with a mean absolute percentage error below 2%.

1.1 ML Algorithms

Initially, a simple linear regression algorithm is tested on datasets with different features. A first dataset is featured only with environmental data, such as min. and max temperature, dew point, air pressure, humidity, visibility, UV index, wind speed, daily light duration, and distinction between working and non-working days, while the second dataset has additional features related to the energy data statistics of the previous day, week and month, collected by the smart meters.

The analysis assessed the influence of the feature (input data) selection on the linear regression model. Hence, the features having better performance with the linear regression are further tested with Gradient-Boosted Trees and Random Forest Regression. Finally, a deep learning algorithm has been implemented to compare the performance of deep learning and non-deep learning algorithms in this use case.

As a summary, following algorithms have been tested:

- Non-deep learning algorithms (classical machine learning):
  - linear regression with only initial features
  - linear regression with additional features
  - Gradient-Boosted Trees (GBT) Regression (for the dataset with additional features)
  - Random Forest Regression (for the dataset with additional features only)
- Deep-learning algorithms:
  - LTMS (for the dataset with additional features only)

1.2 ML Architecture

The results of the best performing model have been compared for discussion with a base case of naïve prediction, which simply assumes the energy consumed in a specific day is equal to the energy consumed the day before.

The analysis has been split into several sequential tasks, which is common for machine learning:

- Initial Data Exploration
- Extract Transform Load
- Feature Engineering
- Model Definition
- Model Training
- Model Evaluation

For a better computational performance, we explored the deployment of the models in a cloud system (IBM cloud system, through Watson Studio execution environment with Python 3.7 and Spark 3.0).

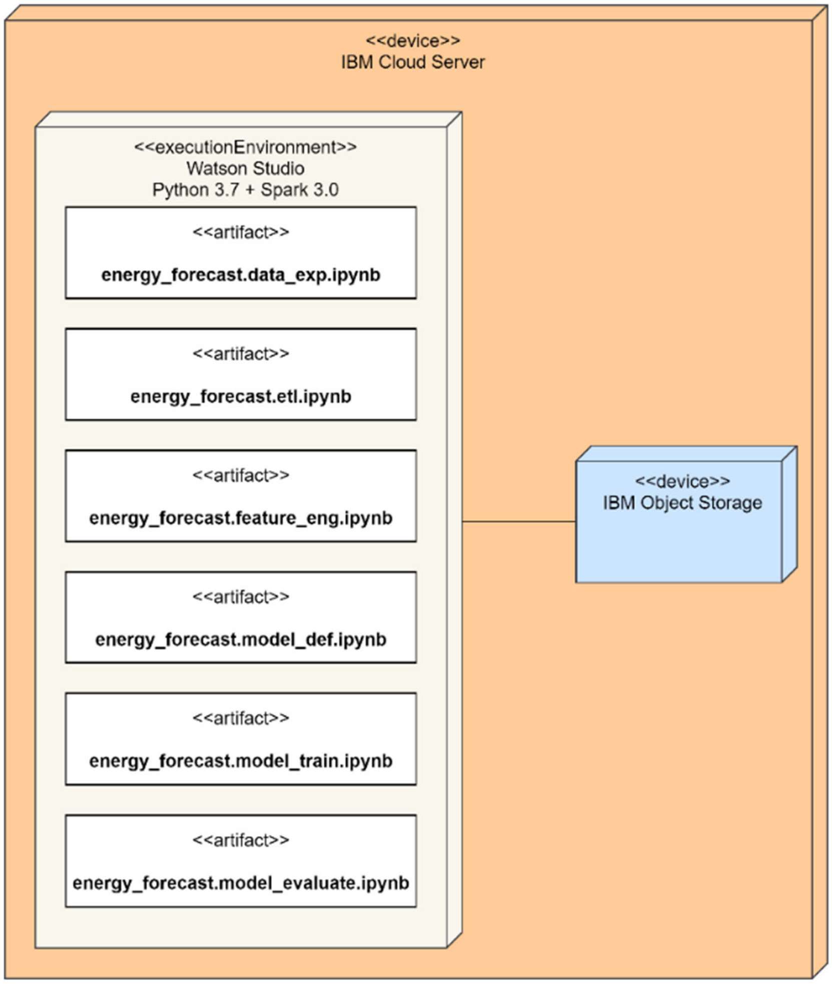


Figure 1 – ML Architecture

1.3 Metrics

The model during training and test has been evaluated by means of R squared metrics, which can be seen as an index showing how well the model fit the data. The R squared move from 0 to 1. Generally, a higher r-squared indicates a better fit for the model.

1.4 Results

The findings of the study in terms of model performance are presented in Table 1 and Figure 2. The best performing models are those with a higher R squared, but also good balance between train and test set results. Models with high r-squared results on training data, but significantly lower performance on test data are overfitting, and they may need to pass through further optimization processes.

	Model	R Square (Train data)	R Square (Test data)
0	Linear Regression (Basic Features)	0.850061	0.682555
1	Linear Regression (Extra Features)	0.934283	0.918921
2	Gradient Boosted Trees (Extra Features)	0.985304	0.884309
3	Random Forest (Extra Features)	0.972449	0.920160
4	Gradient Boosted Trees (Cross Validation)	0.985069	0.910467
5	LSTM	0.977853	0.962497

Table 1 - Comparison of model performance by using R squared metrics (table format).

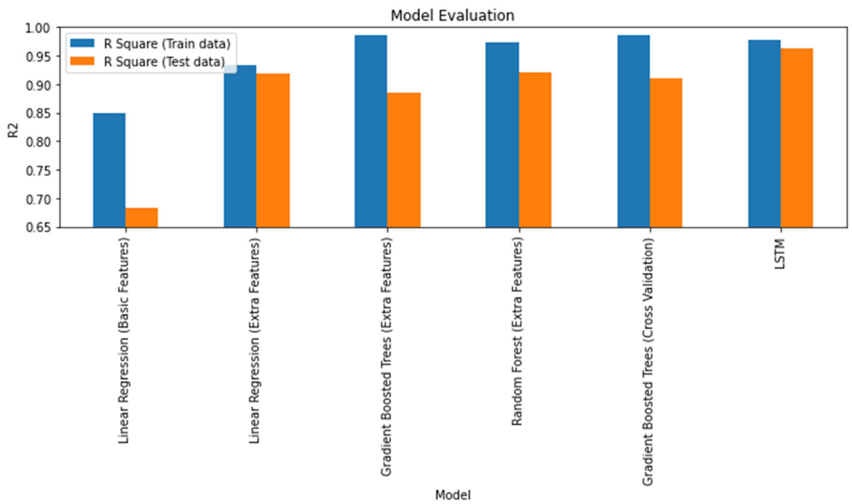


Figure 2 – Comparison of model performance by using R squared metrics (diagram format).

Let's discuss the results of Figure 2. As expected, a good feature selection is very important in classical machine learning. The performance of the model with basic features (only weather data) is quite poor and largely overfitting (first couple of bars – blue/orange – from the left). The addition of extra features, such as statistics of energy consumed on previous day, previous week and previous month, appear to be a good choice. With this additional features, simple linear regression provides a r-squared of about 0.934 on train data and 0.918 on test data.

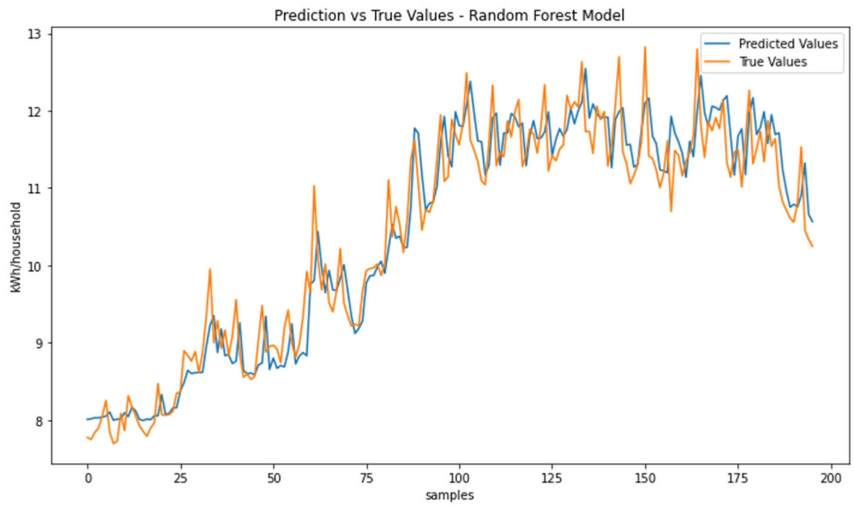
By using the same extra features, two other classical machine learning models have been defined, trained, and evaluated: Gradient Boosted Trees and Random Forest.

Gradient boosted trees scored the highest r-squared on the train data (0.985) but performed quite poorly (0.884) on the test data. Again, the model is overfitting. This time, a technique called cross validation has been used to tune the hyperparameters of the model trying to avoid overfitting. The results of cross validation show an improvement of the r-squared in the test data (0.910), without a significant reduction of performance on the train data. However, the score on test data is still below the random forest, which seems to be the best performing classical machine learning model on the test data.

Finally, a deep learning model has been implemented to compare the results with classical machine learning models. The well-known Long-Short-Term-Memory (LSTM) recurrent neural network has been used, which is particularly suitable for prediction of time series data. This type of neural network can learn the order dependence between items in a sequence, which means that it can contextualize the data for a more accurate prediction.

The LSTM model had the highest R squared on the test dataset and the performance results on train and test data is very well balanced.

The diagram of the predicted value, versus the true values of the test set is presented in Figure 3, for the classical machine learning (random forest) and deep learning (LSTM) models.



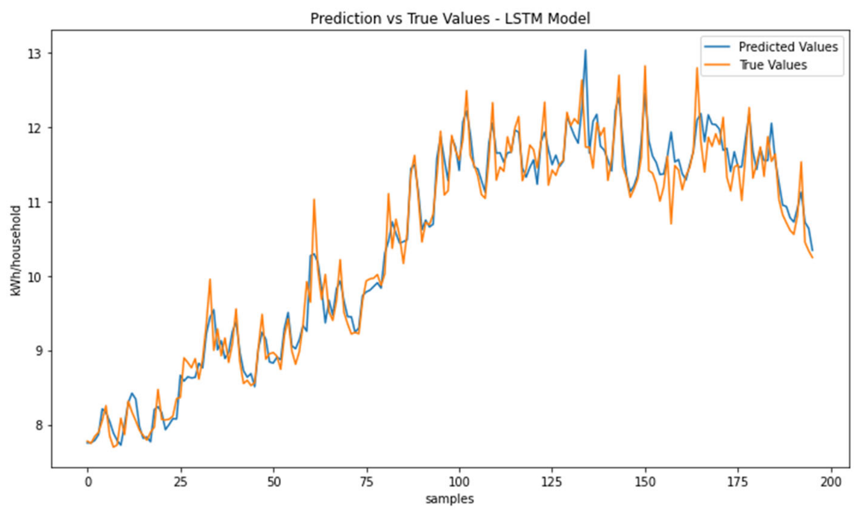


Figure 3 – Prediction vs true values for random forest and LSTM models. The two diagrams may appear very similar, but with attention, we notice several different behaviors in the prediction. LSTM model was the one performing better.

1.5 Further Model Assessment

For the best performing model (LSTM), performance indexes, such as Mean Absolute Error, Mean Squared Error and Mean Absolute Percentage Error have been calculated and plotted in Figure 4.

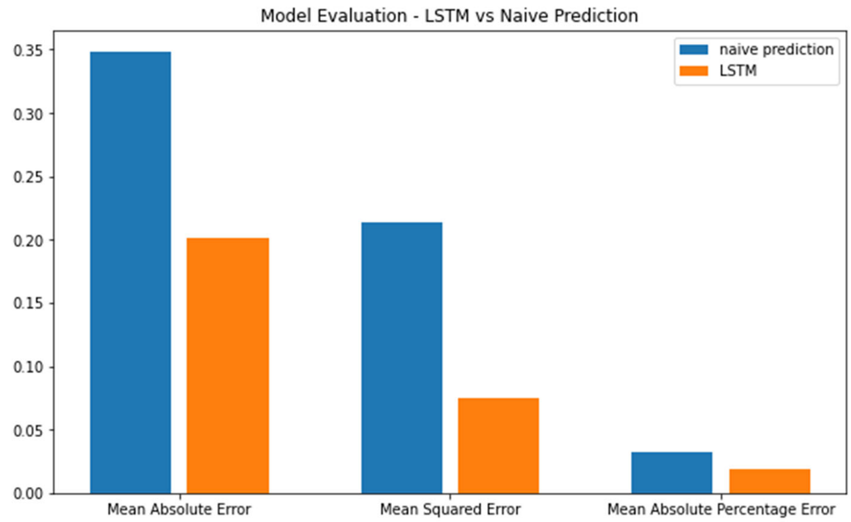


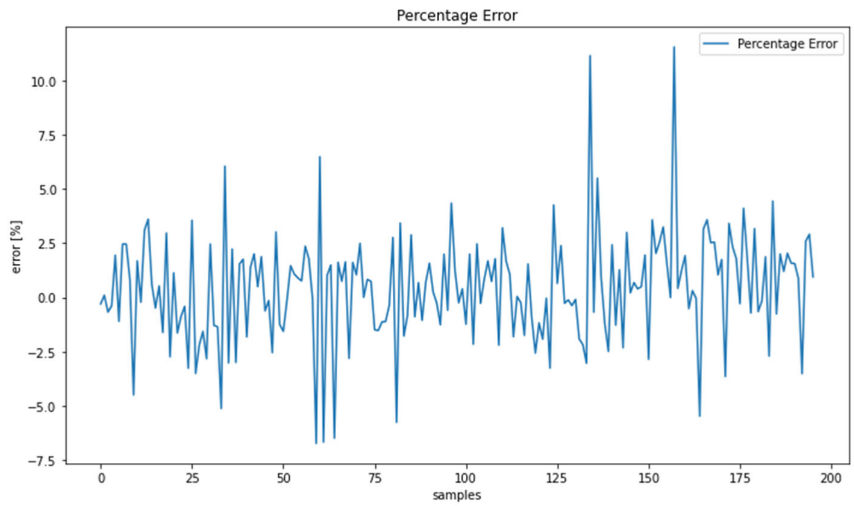
Figure 4 – Performance of LSTM model compared to naïve prediction. A significant improvement is provided by LSTM, compared with naïve prediction, where energy consumption of a specific day is assumed to be equal to the day before.

The LSTM largely outperforms a naïve forecast in all calculated performance indexes. In the case of mean squared error (MSE), the improvement is quite impressive, from about 0.213556 to 0.074922. This means that the implemented LSTM model limits large errors quite well, here penalized by the square of the MSE metric.

Although the mean absolute percentage error is also significantly lower in the LSTM model (from about 3.3% to 1.9%), its improvement is less spectacular. This indicates that variations of energy consumption between two successive days is not very large. Yet, there are enough examples of larger variations to rise the MSE in the naïve forecast. This is where the LSTM brings its major contribution.

Individual point in time with errors above 10% are present even in the LSTM model (see Figure 5).

This reflects the basic principle that train and test data indeed have differences and that the features (smart meter readings and weather data) may not completely describe the observed variable (energy forecast). This does not preclude the usefulness of the model, which can effectively support the energy forecast and the correlated decisions in load and generation management.



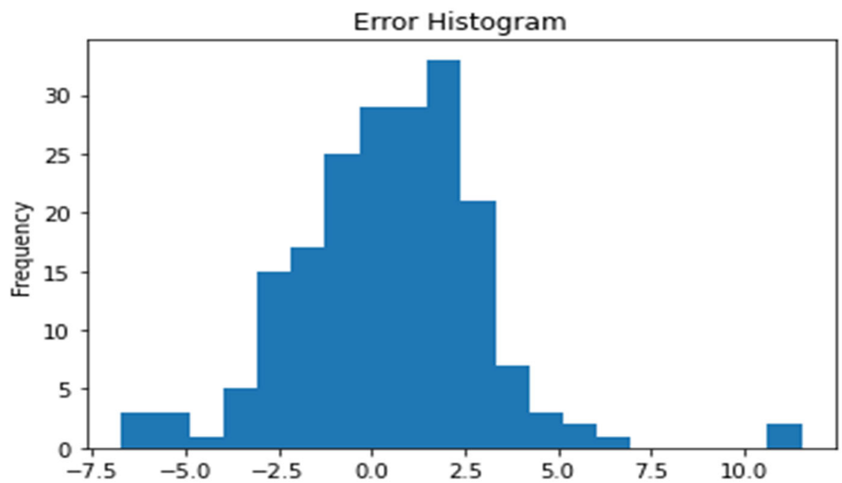


Figure 5 – Percentage Error of LSTM. The mean absolute error is about 1.9%. However, few sample exceptions present much higher errors (above 10%).

1.6 Conclusions

In this use case, we found quite significant improvements of the predictions by using deep learning algorithm, with a target mean absolute percentage error below 2%, even without a full optimization of the model. It is reasonable to assume that the gap between naïve forecast and deep learning model would be even higher in an electricity environment strongly influenced by renewables, electric vehicles, and other stochastic load and generation sources.

About the Author:

Dr. Luca Pizzimbone is a key expert for power system analysis, active in energy transition and evolution of electrical transmission and distribution grids worldwide. He holds a Diploma in Electrical Engineering from the C.I.T.I. G. Galilei of Genova (Italy), a Doctor degree (Laurea di Dottore) in Electrical Engineering from the University of Genoa and he is an IBM certified advanced data scientist.

Being part of the Power Transmission and Distribution Department of Tractebel Engineering GmbH, he has been leading projects in network planning, renewable energy, hybrid systems and smart grids in Africa, Asia, Europe and Middle East.

He is based in Bad Vilbel, Germany.